BAB II

LANDASAN TEORI

2.1 Klasifikasi Teks

Klasifikasi teks ialah proses penempatan teks kesuatu kategori atau kelas sesuai karakteristik dari teks tersebut. Dalam text mining, klasifikasi mengacu kepada aktifitas menganalisis atau mempelajari himpunan teks pre-classified untuk memperoleh suatu model atau fungsi digunakan untuk mengelompokkan teks lain yang belum diketahui kelasnya kedalam satu atau lebih kelas predefined tersebut.

Klasifikasi teks ialah proses pengelompokan dokumen kedalam kelas yang berbeda, tahapannya tiap dokumen menunjuk pada satu kelas tertentu maka dibutuhkan proses untuk menggali informasi dari dokumen tersebut. Sehingga dokumen dapat merepresentasikan kelasnya, sehingga tiap kata yang muncul dalam dokumen mempunyai nilai [6].

Klasifikasi teks atau text categorization secara garis besar terbagi dalam tiga kelompok yaitu yang pertama ialah klasifikasi berbasis statistik. Metode yang digunakan pada kelompok ini adalah Naïve Bayesian, K-Nearest Neighbor, Category Center Vector, Support Vector Machine, dan model Maximum Entropy. Kelompok kedua ialah klasifikasi teks berbasiskan keterkaitan dan metode yang digunakan adalah jaringan syaraf tiruan. Kelompok ketiga adalah klasifikasi teks berbasiskan rule atau aturan. Metode yang digunakan adalah decision tree [7].

2.2 Berita

Berita adalah informasi baru atau informasi mengenai sesuatu yang sedang terjadi, disajikan lewat bentuk cetak, siaran, Internet, atau dari mulut ke mulut kepada orang ketiga atau orang banyak. Dalam buku "Dasar - Dasar Jurnalistik A.M. Hoeta Soehoet. Beliau adalah pendiri sekaligus mantan Rektor Institut Ilmu Sosial dan Ilmu Politik (IISIP) Jakarta [8]:

a. Berita ialah keterangan mengenai peristiwa yang sedang terjadi.

b. Berita bagi seseorang adalah keterangan mengenai suatu peristiwa atau isi pernyataan seseorang yang menurutnya perlu diketahui untuk mewujudkan filsafat hidupnya.
c. Berita suratkabar adalah keterangan peristiwa atau isi pernyataan yang perlu bagi pembacanya untuk mewujudkan filsafat hidupnya.

2.3 Text Mining

"Text mining ialah teknik dapat digunakan untuk melakukan klasifikasi dimana text mining merupakan variasi dari data mining berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar" [2]. Text mining didefinisikan sebagai suatu proses menggali informasi dimana user berinteraksi dengan sekumpulan dokumen menggunakan tools analisis, merupakan komponen-komponen data mining salah satunya ialah kategorisasi. Text mining memiliki tujuan mendapatkan informasi yang berguna dari sekumpulan dokumen Jadi, sumber data digunakan pada text mining merupakan sekumpulan teks memiliki format tidak terstruktur atau minimal semi terstruktur. Tugas text mining ialah *text categorization* dan *text clustering*.

Sistem *text* mining terdiri dari komponen *text preprocessing*, *feature selection*, dan komponen data mining. Komponen *text preprocessing* berfungsi mengubah data tekstual tidak terstruktur seperti dokumen, kedalam data ters truktur dan disimpan kedalam basis data. *Feature selection* memilih kata tepat dan berpengaruh pada proses klasifikasi. Komponen terakhir akan menjalankan teknik data mining pada output dari komponen sebelumnya [9].

2.4 Text Preprocessing

Text Preprocessing merupakan tahapan awal dalam pengolahan teks [1]. tujuan dilakukan tahap preprocessing ini adalah untuk merubah data yang tidak terstruktur menjadi data yang tersetruktur. Tahap *preprocessing* terdiri dari *case folding*, *tokenizing*, *filtering*, *stemming.* Selain itu pemilihan fitur dan pembobotannya merupakan bagian dari tahap ini [10].

2.4.1 Case folding

Case Folding adalah proses pengubahan karakter huruf besar menjadi huruf kecil. Hanya huruf ‘a‘ sampai ‘z‘ yang diterima. Karakter selain huruf akan dihilangkan dan dianggap sebagai delimiter.

2.4.2 Tokenezing

Tokenizing adalah proses memotong sebuah urutan karakter menjadi sebuah potongan-potongan, yang disebut token, mungkin pada waktu yang sama membuang karakter tertentu, seperti tanda baca.

2.4.3 Filtering

Proses filtering melakukan penyaringan kata hasil dari tokenizing, dimana kata yang tidak relevan dibuang. Pada tahap ini tindakan yang dilakukan adalah menghilangkan stopword (stopword removal). Stopword adalah kosakata yang bukan merupakan ciri (kata unik) dari suatu dokumen. Misalnya “di”, “oleh”, “pada”, “sebuah”, “karena” dan lain sebagainya [2].

2.4.4 Stemming

Stemming adalah proses pemotongan berbagai imbuhan atau affixes seperti awalan (prefixes), sisipan (infixes), akhiran (suffixes) dan kombinasi awalan dan akhiran (confixes). Tujuan dari stemming ini adalah mengurangi bentuk jamak atau bentuk turunan yang berhubungan dengan kata dasar yang umum [2].

Pada penelitian ini menggunakan *library* sastrawi stemmer, algoritma yang diterapkan ialah Nazief dan Andriani, kemudian di tingkatkan lagi oleh CS (*Confix Stripping*), ditingkatkan lagi oleh algoritma ECS (*Enhaced Confix Stripping*), dan ditingkatkan lagi oleh *Modified* ECS [11]. Banyak persoalan stemming yang diatasi dengan menggunakan algoritma tersebut, yaitu:

- Mencegah overstemming dengan kamus kata dasar
- Mencegah understemming dengan aturan – aturan tambahan
- Kata bentuk jamak berhasil distem; lari-lari → lari

2.4.5 Dictionary Construction

Proses *dictionary constraction* ialah proses konversi dokumen teks menjadi *vektor fitur*. *Term* yang dimasukan kedalam *vektor fitur* ialah *term*

yang sudah melalui proses *stemming*, setiap fitur didalam vektor berkorespondensi dengan kata pada *dictionary*.

Metode yang digunakan untuk *dictionary construction* ialah *inverted index* atau disebut *inverted file*. *Inverted index* dibagi menjadi dua bagian yaitu *term list* dan *posting list*. *Term - term* yang ada pada dokumen latih disebut *term list*. Setiap term akan tebentuk list yang menyimpan pada dokumen dimana term tersebut muncul. Setiap item dari list yang menyimpan jumlah term muncul pada sebuah dokumen disebut posting, list dari posting-posting disebut posting list atau inverted list.

2.4.6 Feature Selection

walaupun *text* sudah melalui proses *filtering*, tetapi tidak semua kata yang tersisa menggambarkan isi dokumen. Proses *feature setection* bertujuan mengurangi dimensi dari suatu kumpulan teks. Dengan kata lain, menghapus kata - kata yang tidak menggambarkan isi dokumen berdasarkan frekuensi kemunculan kata tersebut [12].

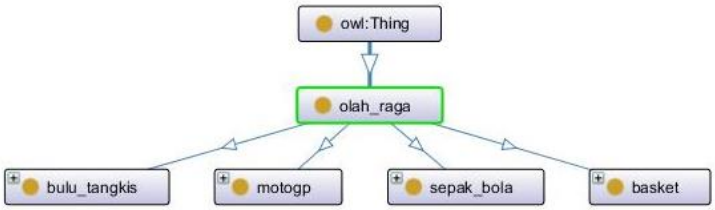
2.4.7 Ontology Extraction

Ontology ialah sebuah deskripsi formal tentang sebuah konsep secara eksplisit dalam sebuah domain dari setiap konsep beserta dengan batasannya. Dalam penelitian ini ontology digunakan untuk menentukan kemiripan makna dari teks pada dokumen tersebut.

Ontologi digunakan untuk berbagi pengetahuan. Ontologi meningkatan informasi dan pengertian. Ontologi memiliki peran penting pada informasi berbasis komputer yang bersifat heterogen. Ontologi memiliki peran penting pada web generasi kedua yang oleh Tim Berners-Lee sebut sebagai “Semantic Web”. Mesin pencari atau search engine akan menemukan halaman dengan kata yang secara sintaksis berbeda tetapi memiliki kesamaan secara semantik [13]. Secara teknis ontologi dapat dipresentasikan dalam bentuk *class, property, facet, dan intances*. *Class* menjelaskan konsep atau makna dari suatu domain.

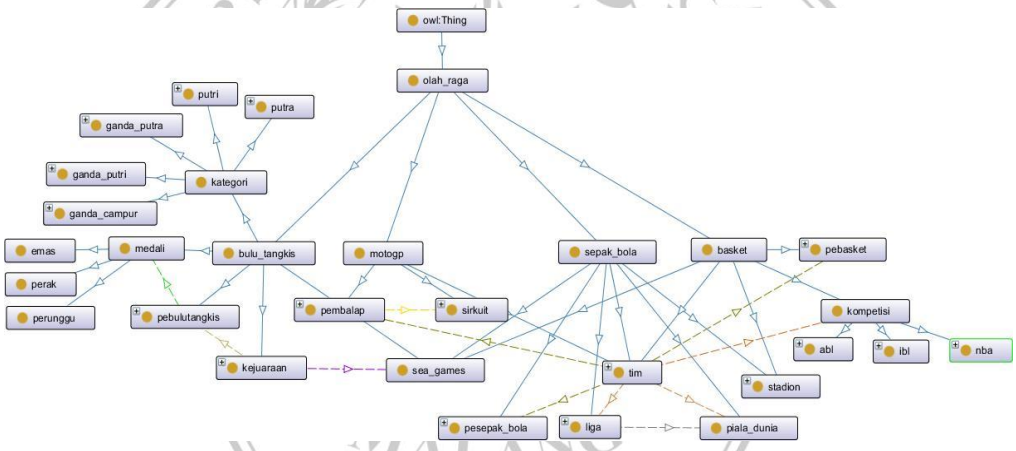
Pemodelan ontologi diawali dengan mendefinisikan root dari ontologi. Root ini diberi nama olahraga, root direpresentasikan sebagai sebuah kelas,

kelas root memiliki 4 subkelas, yaitu bulutangkis, basket, sepak bola, dan motogp. Representasi olahraga dapat dilihat pada gambar 1.



Gambar 1 *representasi* ontologi olahraga

Keempat kategori tersebut juga memiliki subkelas, setiap subkelas dari kelas memiliki property dan instance. Property digunakan untuk mendefinisikan atribut dari subkelas. Selain itu, property juga digunakan untuk mendefinisikan relasi antara subkelas dengan subkelas lainnya. Instance digunakan untuk mendefinisikan objek dari sebuah *property*. Representasi dari masing-masing kelas dapat dilihat pada gambar 2.



Gambar 2 *representasi* domain olahraga pada ontologi

2.4.8 Feature weighting

Feature weighting merupakan proses pembobotan. Metode yang digunakan ialah TF-IDF (*Term Frequency-Inverse Document Frequency*). *Term Frequency* ialah pembobotan *term* berdasarkan perhitungan jumlah t*erm* yang muncul pada suatu dokumen. Semakin tinggi nilai TF (Term frequency) maka semakin penting pula term tersebut untuk mendeskripsikan suatu dokumen. Inverse Document Frequency (IDF) adalah proses mengukur term yang jarang muncul pada corpus. Rumus untuk menentukan nilai IDF adalah sebagai berikut [14] :

$$idfi = \log \frac{N}{dfi} \quad (2\text{-}1)$$

Dengan

N = jumlah keseluruhan dokumen.

dfi = banyaknya dokumen yang mempunyai term i

Sehingga persamaan bobot TF-IDF dapat dituliskan seperti ini :

$$tfidf(i,j) = tf(i,j) \times \log\left(\frac{N}{df(j)}\right) \quad (2\text{-}2)$$

2.5 Data Mining

Data mining adalah bidang multi-disiplin yang menggambarkan kerja dari statistik, teknologi basis data, kecerdasan buatan, pengenalan pola, mesin pembelajaran, teori informasi, pengambilan pengetahuan, pencarian informasi, komputasi tingkat tinggi, dan visualisasi data [15]. Sedangkan dalam buku edisi ke 3 (tiga), ditulis oleh Jiawei Han, Micheline Kambar dan Jian Pei [16], data mining, sering juga disebut sebagai knowledge discovery from data (KDD), adalah ekstraksi otomatis dari sebuah pola yang merepresentasikan pengetahuan implisit yang disimpan atau didapat dalam database yang besar, gudang data, web, tempat penyimpanan informasi yang besar lainnya atau berkas data.

2.6 Decesion Tree

Decision tree ialah diagram alir yang mirip struktur *tree*, dimana setiap *internal node* menotasikan atribut yang di uji, setiap cabangnya merepresentasikan hasil atribut, dan *leaf node* merepresentasikan kelas - kelas tertentu.[3].

2.7.1 Algoritma C5.0

Algoritma C5.0 ialah algoritma data mining, khususnya diterapkan pada teknik *decision tree*. Algoritma C5.0 merupakan penyempurnaan algoritma sebelumnya, dibentuk oleh Ross Quinlan pada tahun 1987, yaitu ID3 dan C4.5. Pemilihan atribut diproses menggunakan *information gain*. Dalam memilih atribut untuk pemecah obyek, beberapa kelas dipilih atribut yang menghasilkan *information gain* paling besar dan menjadi parent bagi node selanjutnya [4].

Rumus *information gain* ialah[3] :

$$I(S_1, S_2, \dots, S_3) = -\sum_{i=1}^{m} p_i \log_2 p_i \quad (2\text{-}3)$$

Keterangan :

S = himpunan kasus

S_1 = jumlah sampel

p_i = proporsi kelas

Untuk mendapatkan informasi nilai subset dari atribut A tersebut maka digunakan rumus sebagai berikut :

$$E(A) = \sum_{j=i}^{y} \frac{S_{1j}+\cdots+S_{mj}}{S} I\left(S_{1j} + \cdots + S_{mj}\right) \quad (2\text{-}4)$$

Keterangan :

$\frac{S_{1j}+\cdots+S_{mj}}{S}$ = jumlah subset j yang dibagi dengan jumlah sampel S

Untuk mendapatkan nilai gain selanjutnya digunakan rumus dibawah ini :

$$Gain\ (A) = I(S_1, S_2, \ldots, S_m) - E(A) \quad (2\text{-}5)$$

Keterangan

A = atribut

S = himpunan kasus

S_1= jumlah sampel

Namun pada kasus klasifikasi teks digunakan perhitungan information gain seperti di bawah ini [17]:

$$I(w) = -\sum_{i=1}^{k} P_i \log(P_i) + F(w).\sum_{i=1}^{k} P_i(w)\log(P_i(w)) + \big(1 + F(w)\big).\sum_{i=1}^{k}(1 - P_i(w))\log(1 - P_i(w)) \quad (2\text{-}6)$$

Keterangan

Pi = probabilitas global dari class i

Pi (w) = probabilitas dari class i dimana dokumen berisi term w

F(w) = fraksi atau jumlah dokumen yang berisi term w

"Fitur algoritma C5.0 lebih unggul dari algoritma terdahulunya dan mengurangi kelemahan algoritma *decision tree* sebelunya. ialah [3]:

a. C5.0 dirancang dapat menganalisis basis data subtansial, berisi puluhan sampai ratusan record dan satuan hingga ratusan field numerik serta nominal.
b. klasifikasi C5.0 disajikan dalam dua bentuk yaitu, menggunakan pohon keputusan dan sekumpulan aturan IF-then.

c. Algoritma C5.0 tidak membutuhkan pengetahuan tinggi tentang statistik atau machine learning.

2.7.2 Splitting Attribute

Dalam kasus klasifikasi teks ini atribut yang digunakan adalah atribut numeric atau continuous, sehingga digunakan metode standart yaitu binary split. Pada atribut numeric atau continuous ini setiap atribut memiliki banyak kemungkinan split-point. *Split-point* yang dipilih adalah *split-point* terbaik dimana *split-point* ini akan menjadi pembatas nilai-nilai pada atribut *A*. Untuk mencari *split-point* tersebut, nilai-nilai pada atribut harus diurutkan dari yang terkecil sampai yang terbesar, lalu nilai tengah dari pasangan nilai yang berdekatan dianggap sebagai salah satu kemungkinan *split-point*. Oleh karena itu, jika atribut *A* memiliki nilai *v*, maka akan ada *v* - 1 banyaknya kemungkinan *split-point* yang akan dievaluasi. Sebagai contoh nilai tengah dari nilai a_i dan a_{i+1} adalah :

$$\frac{a_i+a_{i+1}}{2} \quad (2\text{-}7)$$

Jika nilai-nilai dari *A* diurutkan di awal, kemudian menentukan "*best split*" untuk *A* hanya membutuhkan satu kali melewati nilai-nilai itu. Untuk setiap kemungkinan *split-point* untuk *A*, kita mengevaluasi $Info_A(D)$ dengan menggunakan persamaan :

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} x\ Info(D_j) \quad (2\text{-}8)$$

dimana Info(D_j) sesuai dengan persamaan (2-3).

Evaluasi $Info_A(D)$ dimana jumlah partisi adalah dua, yaitu v = 2 (atau j =1,2). Titik atau point dengan information gain minimum dipilih sebagai split-point untuk atribut A. Setiap nilai A ≤ split-point akan terpartisi ke D_1 dan setiap nilai A > split-point akan terpartisi ke D_2.

2.7 Evaluasi

Penelitian *text categorization* evaluasi dilakukan ketika proses klasifikasi apakah metode digunakan telah mengklasifikasi secara benar. Evaluasi ini biasanya membutuhkan sebuah matrik yang disebut dengan matrix confusion. Matrix confusion berisi informasi tentang klasifikasi yang aktual dan terprediksi

yang dilakukan oleh sistem. Pada penelitian ini metode evaluasi yang digunakan ialah *recall, precision, dan F1-Measure*.

Recall ialah jumlah dokumen terklasifikasi dengan benar oleh sistem dibagi dengan jumlah dokumen yang seharusnya bisa dikenali sistem. *Precision* ialah jumlah dokumen diklasifikasikan dengan benar oleh sistem dibagi dengan jumlah keseluruhan klasifikasi. *F-measure* merupakan nilai yang mewakili kinerja keseluruhan sistem dan merupakan penggabungan nilai *recall* dan *precision* [18].

Persamaan untuk menghitung Precesion, Recall, F-Measure dan

$$Precision = \frac{TP}{TP+FP} \quad (2\text{-}10)$$

$$Recall = \frac{TP}{TP+FN} \quad (2\text{-}11)$$

$$F - Measure = \frac{2\ x\ recall\ x\ precision}{recall+precision} \quad (2\text{-}12)$$

$$Accuracy = \frac{TP+TN}{\text{TP+TN+FP+FN}} \quad (2\text{-}13)$$